# **Московский государственный технический университет имени Н. Э. Баумана.**

Факультет “Радиотехнический”

Кафедра ИУ5 ”Системы обработки информации и управления”

Курс “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе № 2-3.

“Функциональные возможности языка Python”

Выполнила:

Студент группы РТ5-31Б

Губанова В.Е

Подпись и дата:

Проверил:

Преподаватель кафедры ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва,2024 г.

### Текст программы:

### Main.py

import time  
from lab\_python\_fp.field import field  
import lab\_python\_fp.unigue as unique  
import lab\_python\_fp.gen\_random as gen\_random  
import lab\_python\_fp.print\_result as print\_result  
import lab\_python\_fp.cm\_timer as cm\_timer  
def main():  
 #11111111111111111111111111111111111111111111111111111111  
 goods = [  
 {'title': 'Ковер', 'price': 2000, 'color': 'green'},  
 {'title': 'Диван для отдыха', 'color': 'black'}  
 ]  
  
  
 print(list(field(goods, 'title')))  
 print(list(field(goods,'title', 'price')))  
  
 #22222222222222222222222222222222  
 G = gen\_random.gen\_random(10,5,10)  
 for item in G:  
 print(item)  
 print('.........')  
 #333333333333333333333333333333333  
 X=unique.Unique(G, ignore\_case=True)  
  
 for item in X:  
 print(item)  
 print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  
 #4444444444444444444444444444  
 L = [-3, 8, 27, -15.49]  
 result = sorted(L, key=abs, reverse=True)  
 print(result)  
 print('YYYYYYYYYYYYYYYYYYYYY')  
 LL = [28, -45, -1, 0, 55]  
 result = sorted(LL, key=lambda LL: LL if LL >= 0 else -LL, reverse=True)  
 print(result)  
 #555555555555555555555555555555555  
 print\_result.test\_1()  
 print\_result.test\_2()  
 print\_result.test\_3()  
 print\_result.test\_4()  
 #66666666666666666666666666  
 with cm\_timer.cm\_timer\_1():  
 time.sleep(5.5)  
 with cm\_timer.cm\_timer\_2():  
 time.sleep(5.5)  
  
  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Cm\_timer.py

import time  
class cm\_timer\_1:  
  
 def \_\_enter\_\_(self):  
 self.start\_time = time.time()  
 return self  
  
 def \_\_exit\_\_(self, exc\_type, exc\_val, exc\_tb):  
 elapsed\_time = time.time() - self.start\_time  
 print(f'Elapsed time1: {elapsed\_time} seconds')  
from contextlib import contextmanager  
  
@contextmanager  
def cm\_timer\_2():  
 try:  
 start\_time = time.time()  
 yield start\_time  
 finally:  
 elapsed\_time = time.time() - start\_time  
 print(f'Elapsed time2: {elapsed\_time} seconds')

Field.py

def field(items, \*args):  
 assert len(args) > 0  
  
 for item in items:  
 if len(args) == 1:  
 v = item.get(args[0])  
 if v is not None:  
 yield v  
 else:  
 result = {key: item.get(key) for key in args if item.get(key) is not None}  
 if result:  
 yield result

Gen\_random.py

from random import randint  
def gen\_random(num\_count, begin, end):  
 for i in range(num\_count):  
 yield randint(begin, end)

Print\_result.py

def print\_result(func):  
 def wrapper():  
 print(f'{func.\_\_name\_\_}')  
 result = func()  
 if type(result) == list:  
 for item in result:  
 print(item)  
 elif type(result) == dict:  
 for key, value in result.items():  
 print(key, '=', value)  
 else:  
 print(result)  
  
 return result  
 return wrapper  
  
@print\_result  
def test\_1():  
 return 1  
  
@print\_result  
def test\_2():  
 return 'iu5'  
  
@print\_result  
def test\_3():  
 return {'a': 1, 'b': 2}  
  
@print\_result  
def test\_4():  
 return [1, 2]

Unique.py

class Unique(object):  
 def \_\_init\_\_(self, items, \*\*kwargs):  
 self.SORCE=items  
 if len(kwargs) == 0:  
 self.case = False  
 else:  
 self.case=kwargs['ignore\_case']  
 print(self.case)  
  
 self.L = []  
  
  
 def \_\_next\_\_(self):  
 for item in self.SORCE:  
 if isinstance(item, str) and self.case == False:  
 if item.lower() not in self.L:  
 self.L.append(item.lower())  
 return item.lower()  
 else:  
 if item not in self.L:  
 self.L.append(item)  
 return item  
 raise StopIteration  
  
 def \_\_iter\_\_(self):  
 return self

Результат:

